<u>Amendments to the Specification:</u>

Please replace paragraph 23 with the following amended paragraph:

~~Fig. 5 is a diagram~~ <u>Figs. 5A-5B are diagrams</u> of an embodiment of the data structures used by the system of Fig. 2 to store annotation data ~~(on two sheets 5-1 and 5-1)~~;

Please replace paragraph 24 with the following amended paragraph:

Fig. [[5A]] <u>5C</u> is a block diagram of an object properties table data structure and a program mapping table data structure.

Please replace paragraph 38 with the following amended paragraph:

In operation and referring also to [[Fig. 2A]] <u>Figs. 2 and 2A</u>, a designer loads video data 22 from a video source 20 into the authoring tool 24. The video data 22 is also sent from the video source 20 to the video encoder 36 for encoding using, for example, the MPEG standard. Using the authoring tool 24 the designer selects portions of a video image to associate with screen annotations. For example, the designer could select a shirt 2 worn by an actor in the video image and assign annotation data indicating the maker of the shirt 2, its purchase price and the name of a local distributor. Conversely, annotation data may include additional textual information about the object. For example, annotation data in a documentary program could have biographical information about the individual on the screen. The annotation data 5 <u>(Fig. 1D)</u> along with the information about the shape of the shirt 2 and the location of the shirt 2 in the image, which is the mask image, as described below, are stored as data structures 25, 25' in a database 28.

Please replace paragraph 40 with the following amended paragraph:

As described above, this annotation data is also sent to the data packet stream generator 40 for conversion into an encoded data packet stream 27. Time stamp data in the transport

stream [[29]] 29' from the video encoder 36 is also an input signal into the data packet stream generator 40 and is used to synchronize the mask and the annotation data with the image data. The data packet stream generator 40 achieves the synchronization by stepping through a program and associating the timing information of each frame of video with the corresponding mask. Timing information can be any kind of information that allows the synchronization of video and mask information. For example, timing information can be timestamp information as generated by an MPEG encoder, timecode information such as is provided by the SMPTE timecode standard for video, frame numbering information such as a unique identifier for a frame or a sequential number for a frame, the global time of day, and the like. In the present illustration of the invention, timestamp information will be used as an exemplary embodiment.

Please replace paragraph 41 with the following amended paragraph:

The encoded video data from the video encoder 36 is combined with the encoded data packet stream 27 from the data packet stream generator 40 in a multiplexer 44 and the resulting augmented transport stream 46 is an input to a multiplexer system 48. In this illustrative embodiment the multiplexer system 48 is capable of receiving additional transport 29' and augmented transport [[46"]] 46' streams. The transport 29' and augmented transport 46' streams include digitally encoded video, audio, and data streams generated by the system or by other methods known in the art. The output from the multiplexer system 48 is sent to the communications channel 12 for storage and/or broadcast. The broadcast signal is sent to and received by the digital receiver 54. The digital receiver 54 sends the encoded video portion of the multiplexed signal to the television 58 for display. The digital receiver 54 also accepts commands from a viewer, using a handheld remote control unit, to display any annotations that accompany the video images. In one embodiment the digital receiver 54 is also directly in communication with an alternative network connection 56 (Fig. 2).

Please replace paragraph 46 with the following amended paragraph:

As shown in an enlarged view of the mask header 38 in Fig. 2D, the header packet includes information relating to the number of packets carrying mask information, encoding information, timestamp information, visibility word information, and the unique identifier (UID) of the object mapping table associated with the particular mask. UIDs and object mapping tables are discussed below in more detail with respect to [[Fig. 5]] Figs. 5A-5B. Similarly, the first packet for each object begins with a sixteen byte header 45, 45' that contains information that enables the digital receiver 54 to extract, store and manipulate the data in the object packets 43, 43'. Also, as shown in an enlarged view of the object data header 45 in Fig. 2D, the object data header information includes the number of packets carrying data for the particular object, the object's data type, the object's UID, and timestamp related information such as the last instance that the object data is used in the program. The type of data structures employed by the system and the system's use of timestamps is discussed below in more detail with respect to figs. [[5]] 5A-5B, 6, and 7.

Please replace paragraph 55 with the following amended paragraph:

When a viewer begins to interact with the annotation system, the receiver 54 can set a flag that preserves the data required to carry out the interaction with the viewer for so long as the viewer continues the interaction, irrespective of the programmatic material that may be displayed on the video display, and irrespective of a time that the data would be discarded in the absence of the interaction by the viewer. In one embodiment, the receiver 54 sets an "in use bit" for each datum or data structure that appears in a data structure that is providing information to the viewer. A set "in use bit" prevents the receiver 54 from discarding the datum or data structure. When the viewer terminates the interaction, the "in use bit" is reset to zero and the datum or data structure can be discarded when its period of valid use expires. Also present in the data structures of the system but now shown in [[Fig. 5]] Figs. 5A-5B is a expiration timestamp for each data structure by which the system discards the data structure once the time of the program

has passed beyond the expiration timestamp. The discarding process is controlled by a garbage collector 532.

Please replace paragraph 83 with the following amended paragraph:

[[Fig. 5]] <u>Figs. 5A-5B</u> show[[s]] data structures that are used in the invention for storing annotated data information. The data structures store information about the location and/or shape of objects identified in video frames and information that enable viewer interactions with identified objects. In particular [[Fig. 5]] <u>Figs. 5A-5B</u> show[[s]] a frame of video 200 that includes an image of a shirt 205 as a first object, an image of a hat 206 as a second object and an image of a pair of shorts 207 as a third object. In one embodiment, to represent the shape and/or location of these objects, the authoring tool 24 generates a mask 210 which is a two-dimensional pixel array where each pixel has an associated integer value independent of the pixels' color or intensity value. According to different embodiments, a mask represents the location information in various ways including by outlining or highlighting the object (or region of the display), by changing or enhancing a visual effect with which the object (or region) is displayed, by place a graphics in a fixed relation to the object or by placing a number in a fixed relation to the object. In alternative embodiments as indicated above, a mask need not represent the shape and/or location of objects in a video frame but may simply contain graphics and/or textual data.

Please replace paragraph 84 with the following amended paragraph:

In the current illustrative embodiment shown in [[Fig. 5]] <u>Figs. 5A-5B</u>, the system generates a single mask 210 for each frame or video image. A collection of video images sharing common elements and a common camera perspective is defined as a shot. In the illustrative mask 210, there are four identified regions: a background region 212 identified by the integer 0, a shirt region 213 identified by the integer 1, a hat region 214 identified by the integer 2, and a shorts region 215 identified by the integer 3. Those skilled in the art will recognize alternative forms of representing objects could be equally well be used, such as mathematical descriptions of an outline of the image. The mask 210 has associated with it a unique identifier

(UID) 216, a timestamp 218, and a visibility word 219. The UID 216 refers to an object mapping table 217 associated with the particular mask. The timestamp 218 comes from the video encoder 36 and is used by the system to synchronize the masks with the video frames. This synchronization process is described in more detail below with respect to Fig. 6. The visibility word 219 is used by the system to identify those objects in a particular shot that are visible in a particular video frame. Although not shown in [[Fig. 5]] <u>Figs. 5A-5B</u>, all of the other data structures of the system also include an in-use bit as described above.

Please replace paragraph 85 with the following amended paragraph:

The illustrative set of data structures shown in [[Fig. 5]] <u>Figs. 5A-5B</u> that enable viewer interactions with identified objects include: object mapping table 217; object properties tables 220, 220'; primary dialog table 230; dialog tables 250, 250', 250"; selectors 290, 290', 290", action identifiers 257, 257', 257"; style sheet 240; and strings 235, 235', 235", 235"', 256, 256', 256", 259, 259', 259", 259"', 259"", 292, 292', 292", 292"'.

Please replace paragraph 86 with the following amended paragraph:

The object mapping table 217 includes a region number for each of the identified regions 212, 213, 214, 215 in the mask 210 and a corresponding UID [[216]] for each region of interest. For example, in the object mapping table 217, the shirt regions 213 is stored as the integer value "one" and has associated the UID 01234. The UID 01234 points to the object properties table 220. Also in object mapping table 217, the hat region 214 is stored as the integer value two and has associated the UID 10324. The UID 10324 points to the object properties table 220'. The object mapping table begins with the integer one because the default value for the background is zero.

Please replace paragraph 95 with the following amended paragraph:

In operation, when a viewer selects an object and navigates through a series of data structures, the system places each successive data structure used to display information to a

viewer on a stack in the memory 128. For example consider the following viewer interaction supported by the data structures shown in [[Fig. 5]] Figs. 5A-5B. First a viewer selects the hat 214 causing the system to locate the object properties table 220' via the object mapping table 217 and to place the object properties table 220' on the stack. It is implicit in the following discussion that each data structure referenced by the view is placed on the stack.

Please replace paragraph 96 with the following amended paragraph:

Next the system displays a primary dialog table that includes the title 235" and price 235" of the hat and where the style of the information presented to the viewer is controlled by the stylesheet 240.In addition the initial display to the viewer includes a series of choices that are rendered based on the information contained in the selector 290. Based on the selector 290, the system presents the viewer with the choice represented by the strings "Exit" 256, "Buy" 256', and "Save" 256" each of which is respectively referenced by the UIDs 9999, 8888, and 7777. The action identifiers Exit 257' and Save 257 are referenced to the system by the UIDs 1012 and [[1020]] 1010 respectively.

Please replace paragraph 97 with the following amended paragraph:

When the viewer selects the "Buy" string '256, the system uses the dialog table 250, UID 1011, to display the color options to the viewer. In particular, the selector 290' directs the system to display to the viewer the strings "Red" 292, "Blue" 292', "Green" 292", and "Yellow" 292"', UIDs 1111, 2222, 3333, 4444 respectively. The title for the dialog table 250 is located by the system through the variable Symboll 266. When the object properties table 220' was placed on the stack, the Symboll [[266]] 266' was associated with the UID 2001. Therefore, when the system encounters the Symboll 266' it traces up through the stack until it locates the Symboll [[266]] 266' which in turn directs the system to display the string "Pick Color" [[259]] 259' via the UID 2001.

Please replace paragraph 98 with the following amended paragraph:

When the viewer selects "Blue" 2222 string 292', the system executes the action identifier associated with the UID 5555 and displays a dialog table labeled by the string "Pick Size" 259 located through Symbol2, UDI 2002. Based on the selector 290" located by the UID [[2004]] 2003, the system renders the string "Large" 259", UID 1122, as the only size available. If the viewer had selected another color, he would have been directed to the same dialog table, UID 5555, as the hat is only available in large. After the viewer selects the string "Large" 259", the systems presents the viewer with the dialog table 250", UID 6666, to confirm the purchase. The dialog table 250" use the selector 290", UID 2003, to present the viewer the strings "Yes" and "No", UIDs 1113 and 1114 respectively. After the viewer selects the "Yes" string [[259"]] 259''', the system transmits the transaction as directed by the action identifier submit order 257:, UID 1013. Had the viewer chose the "No" strong 259''' in response to the confirmation request, the system would have exited the particular viewer interaction by executing the action identifier exit 257'. As part of the exit operation, the system would have dumped from the stack the object properties table 220' and all of the subsequent data structures placed on the stack based on this particular interaction with the system by the viewer. Similarly, after the execution of the purchase request by the system, it would have dumped the data structures from the stack.

Please replace paragraph 103 with the following amended paragraph:

Referring to Figure [[5A]] 5C, there is shown an object properties table 220" containing a link type field 270 having a corresponding link type entry in the UID field and a stream_num field 227 with a corresponding PID 228. To enable video stream switching, the authoring tool 24 selects the PID 228 corresponding to the PID of a PMT 229 of a particular program stream. When the object corresponding to the object properties table 220" is selected, the digital receiver 54 uses the video link entry 271 of the link type field 270 to determine that the object is a video link object. The digital receiver 54 then replaces the PID of the then current PMT with the PID 228 of the PMT 229. The digital receiver 54 subsequently uses the PMT 229 to extract data

corresponding to the new program. In particular the program referred to by the PMT 229 includes a video stream 260 identified by PID17, two audio streams 261, 262 identified by PID18 and PID19, and a private data stream 263 identified by PID20. In this way the viewer is able to switch between different program streams by selecting the objects associated with those streams.

Please replace paragraph 161 with the following amended paragraph:

~~In step 1166, the~~ <u>The</u> computer makes a selection based on the outcome of the determination performed in step 1164. If there is a positive outcome of the determination step 1164, the computer fills a region that includes the location in the successive two-dimensional section with the selected symbol, as indicated at step 1168. As indicated at step 1170, beginning with the newly-filled region in the successive two-dimensional section, the computer repeats the moving step 1162, the determining step 1164 and the filling step 1168 (that is, the steps recited immediately heretofore) until the determining step results in a negative outcome.